Postel (A)

Network Working Group
Request for Comments: 925

J. Postel
ISI
October 1984

Multi-LAN Address Resolution

STATUS OF THIS MEMO

This memo is prompted by RFC-917 by Jeffery Mogul on "Internet Subnets". In that memo, Mogul makes a case for the use of "explicit subnets" in a multi-LAN environment. In this memo, I attempt to make a case for "transparent subnets". This RFC suggests a proposed protocol for the ARPA-Internet community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

INTRODUCTION

The problem of treating a set of local area networks (LANs) as one Internet network has generated some interest and concern. It is inappropriate to give each LAN within an site a distinct Internet network number. It is desirable to hide the details of the interconnections between the LANs within an site from people, gateways, and hosts outside the site. The question arises on how to best do this, and even how to do it at all. One proposal is to use "explicit subnets" [1]. The explicit subnet scheme is a call to recursively apply the mechanisms the Internet uses to manage networks to the problem of managing LANs within one network. In this note I urge another approach: the use of "transparent subnets" supported by a multi-LAN extension of the Address Resolution Protocol [2].

OVERVIEW

To quickly review the Address Resolution Protocol (ARP). Each host on a broadcast LAN knows both its own physical hardware address (HA) on the LAN and its own Internet Address (IA). When Host-A is given the IA of Host-B and told to send a datagram to it, Host-A must find the HA that corresponds to Host-B's IA. To do this Host-A forms an ARP packet that contains its own HA and IA and the IA of the destination host (Host-B). Host-A broadcasts this ARP packet. The hosts that receive this ARP packet check to see if they are destination sought. If so, they (it should be only Host-B) send a reply specifically addressed to the originator of the query (Host-A) and supplying the HA that was needed. The Host-A now has both the HA and the IA of the destination (Host-B). The Host-A adds this information to a local cache for future use.

Note: The ARP is actually more general purpose than this brief sketch indicates.

The idea in this memo is to extend the ARP to work in an environment of multiple interconnected LANs.

To see how this could work let us imagine a "magic box" (BOX) that is connected as if it were an ordinary host to two (or more) LANs.

Hosts continue to behave exactly as they do with the basic ARP.

When an ARP query is broadcast by any host the BOX reads it (as do all the hosts on that LAN). In addition to checking whether it is the host sought (and replying if it is), the BOX checks its cache of IA:HA address mappings in the cache that it keeps for each LAN it is attached to.

Case 1: If the mapping for the host is found in the cache for the LAN that the query came from, the BOX does not respond (letting the sought host respond for itself).

Case 2: If the mapping for the host is found in the cache for a different LAN than the query came from, the BOX sends a reply giving its own HA on the LAN the query came from. The BOX acts as an agent for the destination host.

Case 3: If the mapping is not found in any of the caches then, the BOX must try to find out the the address, and then respond as in case 1 or 2.

In case 3, the BOX has to do some magic.

The BOX keeps a search list of sought hosts. Each entry includes the IA of the host sought, the interface the ARP was received on, and the source addresses of the original request. When case 3 occurs, the search list is checked. If the sought host is already listed the search is terminated, if not the search is propagated.

To propagate the search, an entry is first made on the search list, then the BOX composes and sends an ARP packet on each of its interfaces except the interface the instigating ARP packet was received on. If a reply is received, the information is entered into the appropriate cache, the entry is deleted from the search list and a response to the search instigating ARP is made as in case 1 or 2. If no reply is received, give up and do nothing -- no response is sent to the instigating host (the entry stays on the search list).

To terminate the search, give up and do nothing -- no response is sent to the instigating host (the entry stays on the search list).

The entries in the caches and the search list must time out.

For every ARP request that is received, the BOX must also put the sending host's IA:HA address mapping into the cache for the LAN it was received on.

THE MULTI-LAN ADDRESS RESOLUTION PROTOCOL

The plan is to use ARP just as it is. The new element is the "magic box" ("ARP-based bridge") that relays the ARP request into neighboring LANs and acts as an agent for relaying datagrams to hosts on other LANs.

The Details

Hosts continue to behave exactly as they do with the basic ARP.

The LANs are connected together by BOXes (computers that are attached to two or more LANs exactly as hosts are attached to LANs). The BOXes implement the following procedure.

Each BOX keeps a table for each LAN it is connected to (or for each LAN interface). Entries in these tables time out, so these tables are caches of recent information. The entries in these caches are the IA:HA address pairs for that LAN.

When an ARP query is broadcast by any host the BOX reads it (as do all the hosts on that LAN). In addition to checking to see if it is the host sought (and replying if it is), the BOX checks its cache of IA:HA address mappings in the table it keeps for each LAN it is attached to.

Case 1: If the mapping for the host is found in the cache for the LAN that the query came from, the BOX does not respond (letting the sought host respond for itself). The time out on this entry is not reinitialized.

Case 2: If the mapping for the host is found in the cache for a different LAN than the query came from, the BOX sends a reply giving its own HA on the LAN the query came from. The time out on this entry is not reinitialized.

In this case the BOX is indicating that it will act as an

agent for the destination host. When an IP datagram arrives at the BOX, the BOX must attempt to forward it using the information in its address mapping caches.

Case 3: If the mapping is not found in any of the caches, then the BOX must try to find out the the address, and then respond as in case 1 or 2. In this case, the BOX has to do some magic.

The BOX keeps a search list of sought (but not yet found) hosts. Each entry includes the IA of the host sought, the interface the ARP was received on, and the source addresses of the original request.

When case 3 occurs, the search list is checked. If the sought host is already listed the search is terminated, if not the search is propagated.

To propagate the search, an entry is first made on the search list, then the BOX composes and sends an ARP packet on each of its interfaces. These ARP requests contain the IA and HA of the BOX and the IA of the sought host, and request the HA of the sought host. If a reply is received to the ARP request, the information is entered into the appropriate cache, the entry is deleted from the search list and a response to the search instigating ARP requests is made as in case 1 or 2 above. If no reply is received, give up and do nothing -- no response is sent to the instigating host (the entry stays on the search list).

Note that the BOX must make a reasonable effort with its ARP requests, if it is normal for ordinary hosts to retry ARP requests five times, then a BOX must also retry it's ARP requests five times.

To terminate the search, give up and do nothing -- no response is sent to the instigating host (the entry stays on the search list).

There is no negative feedback from an ARP request, so there is no way to decide that a search was unsuccessful except by means of a time out.

For every ARP request that is received, the BOX must also put the sending hosts IA:HA address mapping into the cache for the LAN it was received on.

The entries in the caches and the search list must time out.

The search list must be kept and the termination rule followed to avoid an infinite relaying of an ARP request for a host that does not respond. Once a host is listed in the search list, ARP requests will not be relayed. If a host that is down (or otherwise not responding to ARP requests), comes up (or otherwise begins responding to ARP requests) it will still not become available to hosts in other LANs until the search list entry times out.

There are two approaches to this problem: first, to have a relatively short time out on the search list entries; or second, to have the BOX periodically send ARPs for each entry on the search list.

There are several time outs involved in this scheme.

First, the hosts try to get the address resolved using ARP. They may actually make several attempts before giving up if a host is not responding. One must have an good estimate of the length of time that a host may keep trying. Call this time T1.

Second, there is the time that an entry stays on the search list, or the time between BOX generated ARPs to resolve these addresses. Call this time T2.

Note that this time (T2) must be greater than the sum of the T1s for the longest loop of LANs.

Third, there is the time that entries stay in the cache for each LAN. Call this time T3.

The relationship must be T1 < T2 < T3.

One suggestion is that T1 be less than one minute, T2 be ten minutes, and T3 be one hour.

If the environment is very stable, making T3 longer will result in fewer searches (less overhead in ARP traffic). If the environment is very dynamic making T3 shorter will result in more rapid adaptation to the changes.

Another possibility is to restart the timer on the cache entries each time they are referenced, and have a small value for T3. This would result in entries that are frequently used staying in the cache, but infrequently used information being discarded quickly. Unfortunately there is no necessary relationship between frequency of use and correctness. This

method could result in an out-of-date entry persisting in a cache for a very long time if ARP requests for that address mapping were received at just less than the time out period.

When handling regular datagrams, the BOXes must decrement the IP datagram Time-To-Live field (TTL) and update the IP header check sum. If the TTL becomes zero the datagram is discarded (not forwarded).

ARP, as currently defined, will take the most recent information as the best and most up-to-date. In a complicated multi-LAN environment where there are loops in the connectivity it is likely that one will get two (or more) responses to an ARP request for a host on some other LAN. It is probable that the first response will be from the BOX that is the most efficient path.

The one change to the host implementation of ARP that is suggested here is to prevent later responses from replacing the mapping recorded from the first response.

Potential Problems

Bad Cache Entries

If some wrong information get into a cache entry, it will stay there for time T3. The persistence of old information could prevent communication (for a time) if a host changed its IA:HA mapping.

One way to replace bad or out-of-date entries in a cache would be to have the BOXes explicitly interpret a broadcast ARP reply to require an entry with either this IA or HA to be replaced with this new IA:HA mapping. One could have important servers send a broadcast ARP reply when they come up.

Non-ARP Hosts

It seems unrealistic to expect to use both ARP hosts and non-ARP hosts on the same LAN and expect them to communicate. If all the non-ARP hosts are on the same LAN the situation is considered with under the next heading (Non-Broadcast LANs).

Hosts that do not implement ARP must use some other means of address mapping. Either they hold a complete table of all hosts, or they access some such table in a server via some protocol; or they expect to make all routing decisions based on analysis of address fields.

Non-Broadcast LANs

BOXes that are connected to LANs that do not have broadcast capability and/or LANs where the hosts do not respond to ARP may have a static or dynamic table of the IA:HA mappings for that LAN (or the addresses may be computed from one another). All the hosts on that LAN must be in the table.

When a BOX must find the address mapping and would otherwise send an ARP request into a non-broadcast LAN (this can only happen when the sought host is not the non-broadcast LAN since all the hosts are in the table), it must instead send an ARP type request specifically to each of the other BOXes on that LAN.

Size of Tables

The worst case of the size of the tables in the BOXes is the number of hosts in the set of LANs for each table. That is, the table kept for each LAN interface may (in the worst case) grow to have an entry for each host in the entire set of LANs. However, these tables are really caches of the entries needed for current communication activity and the typical case will be far from the worst case. Most hosts will communicate mostly with other hosts on their own LAN and with a few hosts on other LANs. Most communication on LANs is between work station hosts and server hosts. It can be expected that there will be frequent communication involving the main server hosts and that these server hosts will be entered in the tables of most of the BOXes most of the time.

Infinite Transmission Loops

The possibility of infinite transmission loops through an interconnected set of LANs is prevented by keeping search lists in the BOXes and terminating the search when a request is received for an address already on the list.

Transmission loops of regular datagrams can not persist because them the BOXes must decrement the TTL, and discard the datagram if the TTL is reduced to zero. For debugging purposes it would be useful for a BOX to report to the implementer any datagrams discarded for this reason.

Broadcast

Note that broadcast does not really have anything to do with either transparent subnets or explicit subnets. Since it was discussed in [1], it will be discussed here, too. Two of the three broadcast functions suggested in [1] work just the same and have the same effects, the third can be supported, too.

It is also argued that the support for a broadcast interpretation of IAs is a bigger issue that the question of explicit subnets versus transparent subnets and it should be decided separately.

It is also suggested that broadcast is not really what is desired, but rather multicast is the better function. It may make sense to understand how to do an Internet multicast before adopting a broadcast scheme.

This IP Network

If the IA of this network number and an all ones host number (e.g., 36.255.255.255) is used, an IP level broadcast to all hosts on this Network (all LANs) is intended. A BOX must forward this datagram. A BOX must examine the datagram for potential significance to the BOX itself.

To prevent infinite transmission loops each BOX must keep a list of recent broadcasts. The entries in this list contain the source IA and the Identification field from the datagram header. If a broadcast is received and matches an entry on the list it is discarded and not forwarded. The entries on this list time out in time T2.

This LAN Only

If the IA of all ones (i.e., 255.255.255.255) is used an IP level broadcast to all hosts on this LAN only is intended. A BOX must not forward this datagram. A BOX must examine the datagram for potential significance to the BOX itself.

Another LAN Only

Since the LANs are not individually identified in the IA this can not be supported in the same way. Some have also argued that this is a silly capability to provide.

One way to provide it is to establish a specific IA for each

LAN that means "broadcast on this LAN". For example, 36.255.255.128 means broadcast on LAN A, and 36.255.255.187 means broadcast on LAN B, etc. These addresses would be specially interpreted by the BOXes attached to the specific LAN where they had the special interpretation, other BOXes would treat these address as any other IAs. Where these addresses are specially interpreted they are converted to the broadcast on this LAN only address.

DISCUSSION

The claim for the extended ARP scheme is that the average host need not even know it is in a multi-LAN environment.

If a host took the trouble to analyze its local cache of IA:AH address mappings it might discover that several of the IAs mapped to the same HA. And if it took timing measurements it might discover that some hosts responded with less delay that others. And further, it might be able to find a correlation between these discoveries. But few hosts would take the trouble.

Address Structure

In the explicit subnet scheme, some IA bits are devoted to identifying the subnet (i.e., the LAN). The address is broken up into network, subnet, and host fields. Generally, when fields are use the density of the assigned addresses in the address space goes down. That is, there is a less efficient use of the address space. Significant implementation problems may arise if more subnets than planned are installed and it becomes necessary to change the size of the subnet field. It seems totally impractical to use the explicit subnet scheme with a class C IA.

In the extended ARP scheme the address is simply the network, and host fields. The extended ARP scheme may be used with any class of IA.

Relocating Hosts

In the explicit subnet scheme when a host is unplugged from one LAN and plugged into another its IA must change.

In the extended ARP scheme it may keep the same IA.

One view of the situation suggests that there are really two problems:

1. How does the host discover if the destination is in this LAN or some other LAN?

   This question assumes that a host should know the difference and should do something different in the two cases, and further that once the host knows the answer it also know how to send the data (e.g., directly to the host, or to the box).

      The claim here is that the hosts should not know the difference and should always do the same thing.

2. How do the BOXes that connect LANs know which BOXes are the routes to which LANs?

   This question assumes that the BOXes need some kind of topological knowledge, and exchange BOX-to-BOX protocol information about connectivity.

      The claim here is that the BOXes do not need topological knowledge and do not need to explicitly know about the existence of other BOXes.

It has been suggested that there are two problems: first, how the hosts do routing; and second, how the BOXes do routing. A claim has been made that the competing strategies each have an approach to each problems and one could select a solution made up partly from one approach and partly from another.

   For example: use ARP within the LAN and have the BOX send ARP replies and act as a agent (as in the extended ARP scheme), but use a BOX-to-BOX protocol to get the "which hosts are where" information into the BOXes (as in the explicit subnet scheme).

There are two places where code is involved: a large number of hosts, and a small number of BOXes. In considering the trade off between explicit subnet scheme and extended ARP scheme, the work done in the hosts should weigh a lot more than the work done in the BOXes.

   What do hosts do?

      Explicit Subnet Scheme

         The host must be able to decide if this IA is on this LAN or

some other LAN. If on this LAN then use some procedure to find the HA. If on some other LAN then use some procedure to find the HA of a BOX.

Extended ARP Scheme

In every case the host uses ARP to get a IA:HA mapping.

What do the BOXes do?

Explicit Subnet Scheme

The BOX must be able to decide which LAN within the site the destination host is on. The BOXes must have some routing table that tells for each LAN in the site which interface to send datagrams on. This routing table must be kept up to date, probably by a BOX-to-BOX protocol much like the Internet Gateway-to-Gateway protocol.

Extended ARP Scheme

The BOX must keep caches for each LAN it is attached to of IA:HA mappings, and it must keep a search list. It does not run any BOX-to-BOX protocol, It does not even know if any other BOXes exist.

Topology and Implementation Complexity

Trees

If the organization of the LANs and the BOXes is tree structured, the BOXes may be very simple, they don't have to keep the search lists at all, since there won't be any loops for the ARP-request to traverse.

Loops

If the organization has loops then the search lists are essential. If the topology is kept balanced so that there are no long loops (all loops are about the same size), and the LANs are reasonably compatible in delay characteristics, then the procedure described here will work well.

Complex

If the organization is very complex, topologically unbalanced,

and/or composed of mix of different types of LANS with vastly different delay characteristics, then it may be better to use a BOX-to-BOX routing protocol.

SUMMARY

It would be useful if the Internet community could come to some agreement on a solution to the multi-LAN network problem and could with a unified voice urge work station manufacturers to provide that solution built in.

I urge consideration of the extended ARP scheme expounded on here.

I think that most work stations will be connected to LANs that have a broadcast capability. I think that most work stations will be used in situations that do not require explicit subnets, and most will be used in situations where a class C Internet addresses would be appropriate (and explicit subnets impossible). Thus, i think it would be best to ask manufacturers to include support for ARP in work stations off the shelf. I also think we ought to get busy and create, develop, test, and produce the magic boxes I suggest so that they too are available off the shelf.

Please note that neither this note nor [1] proposes a specific routing procedure or BOX-to-BOX protocol. This is because such a routing procedure is a very hard problem. The plan proposed here will let us get started on using multi-LAN environments in a reasonable way. If we later decide on a routing procedure to be used between the BOXes we can redo the BOXes without having to redo the hosts.

GLOSSARY

ARP

Address Resolution Protocol (see [2]).

BOX

Magic Box. A box (computer) connected to two or more LANs of the same Network. Also called an "ARP-based bridge".

Bridge

A node (computer) connected to two or more administratively indistinguishable but physically distinct subnets, that automatically forwards datagrams when necessary, but whose existence is not know to other hosts. Also called a "software repeater".

Datagram

The unit of communication at the IP level.

Explicit Subnet

A Subnet explicitly identified in the the Internet Address by a subnet address field, and so visible to others both in side and out side the Network.

Gateway

A node (computer) connected to two or more administratively distinct networks and/or subnets, to which hosts send datagrams to be forwarded.

HA

Hardware Address, the address used in a packet on a LAN.

Host Number

The address of a host within an Network, the low-order part of an IA.

IA

Internet Address, as defined in IP.

Internet

The collection of connected Internet Networks (also known as the Catenet). A set of interconnected networks using IP.

IP

Internet Protocol (see [3]).

LAN

Local Area Network.

Multi-LAN Network

A set of LANs treated as one Network, i.e., using one Network Number in common. The individual LANs may be either Explicit Subnets or Transparent Subnets.

Network

A single Internet Network (possibly divided into subnets or composed of multiple LANs), identified by an individual Network Number.

Network Number

An IP Network Number, the high-order part of an IA.

Packet

The unit of communication at the LAN hardware level.

Subnet

A subnet of Network. A portion of a Network (either logical or physical).

Transparent Subnet

A Subnet not identified in the Internet Address, and so invisible to others, (see Multi-LAN Network).

TTL

The IP Time-To-Live field.

REFERENCES

[1] J. Mogul, "Internet Subnets", RFC-917, Stanford University, October 1984.

[2] D. Plummer, "An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware", RFC-826, Symbolics, November 1982.

[3] J. Postel, "Internet Protocol", RFC-791, USC-ISI, September 1981.